

Graph concatenation for quantum codes

Salman Beigi,¹ Isaac Chuang,² Markus Grassl,³ Peter Shor,⁴ and Bei Zeng^{5,6,a)}

¹*Institute for Quantum Information, California Institute of Technology, Pasadena, California 91125, USA*

²*Department of Physics and Department of Electric Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

³*Centre for Quantum Technologies, National University of Singapore, Singapore*

⁴*Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

⁵*Department of Mathematics and Statistics, University of Guelph, Guelph, Ontario N1G2W1, Canada*

⁶*Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario N2L3G1, Canada*

(Received 3 February 2010; accepted 15 December 2010; published online 16 February 2011)

Graphs are closely related to quantum error-correcting codes: every stabilizer code is locally equivalent to a graph code and every codeword stabilized code can be described by a graph and a classical code. For the construction of good quantum codes of relatively large block length, concatenated quantum codes and their generalizations play an important role. We develop a systematic method for constructing concatenated quantum codes based on “graph concatenation,” where graphs representing the inner and outer codes are concatenated via a simple graph operation called “generalized local complementation.” Our method applies to both binary and nonbinary concatenated quantum codes as well as their generalizations. © 2011 American Institute of Physics. [doi:10.1063/1.3534799]

I. INTRODUCTION

The discovery of quantum error-correcting codes (QECCs) and the theory of fault-tolerant quantum computation (FTQC) have greatly improved the long-term prospects for quantum communication and computation technology. This general QECC-FTQC framework leads to a remarkable threshold theorem, which indicates that noise likely poses no fundamental barrier to the performance of large-scale quantum computations.¹

Stabilizer codes, a quantum analog of classical additive codes, are the most important class of QECCs.^{2,3} These codes have dominated the study of QECC-FTQC for the past 10 years because of their simple construction based on Abelian groups. The recently introduced codeword stabilized (CWS) quantum codes framework^{4–6} provides a unified way of constructing a larger class of quantum codes, both stabilizer and nonadditive codes. Based on the CWS framework, many nonadditive codes, which outperform stabilizer codes in terms of coding parameters, have been constructed.

Graphs are closely related to QECCs. It has been shown that every stabilizer code is local Clifford equivalent to a graph code.^{7,8} The basic ingredients of a graph code are a graph and a finite Abelian group from which the code can explicitly be obtained.⁹ Every CWS code also has a canonical form, where it can be fully characterized by a graph and a classical code.^{4,5} When the classical code is linear, the code is a graph code; therefore, graph codes, and hence stabilizer codes, are special cases of CWS codes.

For the construction of good QECCs of relatively large block length and good asymptotical performance, concatenated quantum codes and their generalizations play an important role.^{1,2,10,11}

a) Address all correspondence should be addressed. Electronic mail: zengbei@gmail.com.

Combined with the CWS framework, families of good quantum codes, both stabilizer and nonadditive, have been constructed.^{10,11} Concatenated quantum codes also play a central role in FTQC and the proof of the threshold theorem.^{1,12–15} Given the intimate relations between graphs and quantum codes, a question that arises naturally is whether there is a graphical description for concatenated quantum codes and their generalizations. Moreover, if there were such a description, for the case where both the inner and outer codes are CWS codes, the next question is whether the corresponding graph captures the “quantum nature” of the concatenated code.

Previously, some related results on graph codes have been obtained. For instance, concatenation of graph codes may be described graphically by adding some auxiliary vertices. However, it remains unclear what the final graph after removing those auxiliary vertices will look like.¹⁶ The known examples of generalized concatenated codes only provide graphical descriptions in the case where the outer code is of a special form.^{10,11} However, none of these previous works provide a general systematic graphical description for constructing concatenated quantum codes. Lack of such a description seems to indicate that using graphs to describe quantum codes was a very restricted approach. This issue will be addressed in the present work by developing a systematic method for constructing concatenated quantum codes based on a graph operation called “graph concatenation.”

To state our result more precisely, let us fix some notations. As already mentioned, a CWS code in the canonical form is determined by a graph \mathcal{G} and a classical code \mathcal{C} . If \mathcal{G} is a graph with n vertices and \mathcal{C} is a classical code which encodes k bits into n -bit strings, then $\mathcal{Q} = (\mathcal{G}, \mathcal{C})$ is a CWS code, encoding k qubits into n qubits: if $\alpha_1 \dots \alpha_n$ is the codeword corresponding to the k -bit string r , then the logical state $|r\rangle_L$ is $(Z^{\alpha_1} \otimes \dots \otimes Z^{\alpha_n})|\psi\rangle_{\mathcal{G}}$, where $|\psi\rangle_{\mathcal{G}}$ denotes the graph state corresponding to the graph \mathcal{G} (later we slightly change this definition). The definition in the nonbinary case is similar. Such a code has parameters $[[n, k, d]]_p$, where d is the distance of the code and in the binary case $p = 2$. In general, if \mathcal{C} (is not linear and) contains K codewords, the code is denoted by $((n, K, d))_p$ (in the above case we have $K = 2^k$).

For a concatenated QECC, let $\mathcal{Q}_{\text{in}} = (\mathcal{G}_{\text{in}}, \mathcal{C}_{\text{in}})$ and $\mathcal{Q}_{\text{out}} = (\mathcal{G}_{\text{out}}, \mathcal{C}_{\text{out}})$ denote the inner and outer codes, respectively. We require \mathcal{C}_{in} to be linear, but \mathcal{C}_{out} can be either linear or nonlinear. Since \mathcal{C}_{in} is linear, \mathcal{Q}_{in} is a graph (stabilizer) code. We can then denote the parameters of \mathcal{Q}_{in} by $[[n, k, d]]_p$. For simplicity, throughout the paper we assume that p is a prime number. When \mathcal{Q}_{in} encodes k qubits, the corresponding outer code \mathcal{Q}_{out} of length n' must be a subspace of the Hilbert space $\mathcal{H}_{p^k}^{\otimes n'}$, so the parameters of \mathcal{Q}_{out} are of the form $((n', K', d'))_{p^k}$. Now our problem is to compute the concatenated code which we denote by

$$\mathcal{Q}_c = \mathcal{Q}_{\text{in}} \sqsubset \mathcal{Q}_{\text{out}}. \quad (1)$$

Main Result: *The concatenated quantum code \mathcal{Q}_c can also be described as a CWS code. Indeed,*

$$\mathcal{Q}_c = (\mathcal{G}_c, \mathcal{C}_c) = (\mathcal{G}_{\text{in}} \sqsubset \mathcal{G}_{\text{out}}, \mathcal{C}_{\text{in}} \sqsubset \mathcal{C}_{\text{out}}), \quad (2)$$

where $\mathcal{C}_c = \mathcal{C}_{\text{in}} \sqsubset \mathcal{C}_{\text{out}}$ is the usual concatenation of classical codes and $\mathcal{G}_{\text{in}} \sqsubset \mathcal{G}_{\text{out}}$ is the concatenation of two graphs \mathcal{G}_{in} and \mathcal{G}_{out} which will be defined later. The concatenated graph \mathcal{G}_c can be obtained via a simple graph operation called “generalized local complementation.”

We postpone the description of the process of obtaining the concatenated graph $\mathcal{G}_c = \mathcal{G}_{\text{in}} \sqsubset \mathcal{G}_{\text{out}}$, but let us explain the definition of the generalized local complementation operation in the binary case. Let i be a vertex in a graph \mathcal{G} and denote the set of its adjacent vertices by $N(i)$. Furthermore, let S be a subset of vertices disjoint from $N(i)$. Then the generalized local complementation on i with respect to S is the operation of replacing the bipartite subgraph induced on $N(i) \cup S$ with its complement. The concatenated graph \mathcal{G}_c is obtained by starting from some copies of \mathcal{G}_{out} and \mathcal{G}_{in} and applying a sequence of generalized local complementations.

The main advantage of constructing concatenated quantum codes via Eq. (2) is that the “quantum part” of this construction is fully characterized by the graph concatenation $\mathcal{G}_{\text{in}} \sqsubset \mathcal{G}_{\text{out}}$. Providing the rule for performing this graph concatenation, the problem of constructing concatenated quantum codes becomes purely classical, i.e., constructing the classical concatenated code $\mathcal{C}_{\text{in}} \sqsubset \mathcal{C}_{\text{out}}$. Despite the restriction that \mathcal{C}_{in} must be linear, our method of graph concatenation can be applied to very general situations: both binary and nonbinary concatenated quantum codes and their generalizations.

This paper is organized as follows. In Sec. II, we review definitions of graph states, CWS codes, and graph codes and explain a simple example of concatenation of CWS codes. In Sec. III, for a simple case that the inner code encodes only a single qupit (i.e., $k = 1$) and the outer code is also a graph code, we provide a description of graph concatenation based on the algebraic structure of stabilizers. We prove our main result in Sec. IV. In Sec. V, we discuss the application of our main result to the situation of the generalized concatenated quantum codes (GCQCs). A final discussion and conclusions are given in Sec. VI.

II. GRAPH STATES, CWS CODES, AND GRAPH CODES

In this section we review the stabilizer formalism to fix our notation especially in the nonbinary case, then define CWS codes and graph codes, and finally describe their encoding circuits.

In the following we use lowercase bold letters $\mathbf{a}, \mathbf{b}, \dots$ to represent row vectors. Moreover, $\mathbf{0}_{m \times n}$ denotes the zero matrix of size $m \times n$ and $\mathbf{0}_n = \mathbf{0}_{n \times n}$. Often, when there is no confusion, the subscript in $\mathbf{0}_{m \times n}$ is eliminated.

A. The generalized Pauli group

Let p be a prime number and \mathbb{F}_p be the field of p elements. A qupit is a p -level quantum system whose Hilbert space is represented by the orthonormal basis $\{|r\rangle : r \in \mathbb{F}_p\} = \{|0\rangle, |1\rangle, \dots, |p-1\rangle\}$. Let $\omega = e^{2\pi i/p}$ be a p th root of unity. The (generalized) Pauli matrices X and Z are defined as follows.

$$X|r\rangle = |r+1 \bmod p\rangle, \quad (3)$$

$$Z|r\rangle = \omega^r |r\rangle. \quad (4)$$

It is clear that $X^p = Z^p = I$, and then we can consider the operators X^a and Z^b where $a, b \in \mathbb{F}_p$. We have $Z^b X^a = \omega^{ab} X^a Z^b$; therefore, $X^a Z^b$ and $X^{a'} Z^{b'}$ commute if and only if (iff) $ab' - ba' = 0$ (see, e.g., Ref. 17 for more details).

The group generated by the (generalized) Pauli matrices X and Z is $\{\omega^c X^a Z^b : a, b, c \in \mathbb{F}_p\}$ and is called the (generalized) Pauli group. Note that in the binary case ($p = 2$) the Pauli group is generated by Pauli matrices σ_x and σ_z together with iI ($i = \sqrt{-1}$).

Let n be an arbitrary positive integer. For vectors $\mathbf{a} = (a_1, \dots, a_n)$ and $\mathbf{b} = (b_1, \dots, b_n)$ in \mathbb{F}_p^n , define

$$X^{\mathbf{a}} = X^{a_1} \otimes \dots \otimes X^{a_n} \quad (5)$$

and

$$Z^{\mathbf{b}} = Z^{b_1} \otimes \dots \otimes Z^{b_n}. \quad (6)$$

Again, two Pauli matrices $X^{\mathbf{a}} Z^{\mathbf{b}}$ and $X^{\mathbf{a}'} Z^{\mathbf{b}'}$ commute if and only if $\mathbf{a}\mathbf{b}' - \mathbf{a}'\mathbf{b} = 0$, where $\mathbf{c}\mathbf{d} = c_1 d_1 + \dots + c_n d_n$ is the usual inner product on \mathbb{F}_p^n .

For simplicity, a Pauli operator $X^{\mathbf{a}} Z^{\mathbf{b}}$ is denoted by the vector $(\mathbf{a} | \mathbf{b})$ of length $2n$. Thus two Pauli operators $X^{\mathbf{a}} Z^{\mathbf{b}}$ and $X^{\mathbf{a}'} Z^{\mathbf{b}'}$ commute if and only if their corresponding vectors are orthogonal with respect to the “symplectic inner product” defined by

$$(\mathbf{a} | \mathbf{b}) * (\mathbf{a}' | \mathbf{b}') = \mathbf{a}\mathbf{b}' - \mathbf{a}'\mathbf{b}. \quad (7)$$

B. Stabilizer states

It is easy to see that for a Pauli matrix $g = \omega^c X^a Z^b$, $g^p = I$. (In the case $p = 2$, the statement might only be true after replacing g by ig in order to get $g^2 = I$. Having this in mind, there is no true difference between the binary and nonbinary case in the rest of the paper.) Therefore, the eigenvalues

of g are all p th roots of unity. In fact, if $(\mathbf{a} | \mathbf{b})$ is nonzero, then for any i , ω^i is an eigenvalue of g , and the multiplicity of each of these p eigenvalues is equal to p^{n-1} .¹⁷

Now suppose $g_1 = \omega^{c_1} X^{\mathbf{a}^1} Z^{\mathbf{b}^1}, \dots, g_k = \omega^{c_k} X^{\mathbf{a}^k} Z^{\mathbf{b}^k}$ are k Pauli matrices which pairwise commute and such that the subgroup generated by any $k-1$ of them does not contain the other one. Additionally, we require that the group generated by g_1, \dots, g_k does not contain a nontrivial multiple of identity. Since g_1, \dots, g_k commute, they can be diagonalized simultaneously.

Lemma 1: The common eigenspace of all g_i 's with eigenvalue 1 is a p^{n-k} -dimensional subspace.

This lemma is a well-known fact in the binary case,¹ and a proof for the nonbinary case can be found in Ref. 17.

Representing the operators g_1, \dots, g_k by the vectors of length $2n$, we obtain the $k \times (2n)$ matrix

$$M = \left(\begin{array}{c|c} \mathbf{a}^1 & \mathbf{b}^1 \\ \mathbf{a}^2 & \mathbf{b}^2 \\ \vdots & \vdots \\ \mathbf{a}^k & \mathbf{b}^k \end{array} \right) \quad (8)$$

of rank k (because g_i is not in the subgroup generated by the rest of g_j 's). The rows of M are mutually orthogonal with respect to the symplectic inner product [see Eq. (7)].

If we consider n generators, or equivalently an $n \times (2n)$ full-rank self-orthogonal matrix M , Lemma 1 implies that the common eigenspace of all g_i 's with eigenvalue 1 is a one-dimensional subspace. Hence there is a unique (up to a scalar) nonzero vector $|\psi\rangle$ such that $g_i|\psi\rangle = |\psi\rangle$. In fact, if we consider the group \mathcal{S} generated by the g_i 's, for any $h \in \mathcal{S}$ we have $h|\psi\rangle = |\psi\rangle$. The group \mathcal{S} , which is a maximal Abelian subgroup of the Pauli group modulo its center, is called a *stabilizer group*, and the state $|\psi\rangle$ is called a *stabilizer state*.

Note that for a stabilizer state $|\psi\rangle$, its stabilizer group \mathcal{S} is unique; however, $\{g_1, \dots, g_n\}$ is just some set of generators of \mathcal{S} . Suppose $\{h_1, \dots, h_n\}$ is another generating set of \mathcal{S} . Then for any i there is $u_{ij} \in \mathbb{F}_p$ such that $h_i = g_1^{u_{i1}} \dots g_n^{u_{in}}$. As a result, the vector corresponding to h_i is equal to $(u_{i1}, \dots, u_{in})M$.

Lemma 2: Any set of generators of the stabilizer group \mathcal{S} with k generators can be represented by a matrix UM , where U is an invertible $k \times k$ matrix.

C. Clifford group

The Clifford group is the normalizer of the Pauli group. In the binary case, it is well-known that the Clifford group is generated by the Hadamard gate, the phase gate, and the controlled-NOT gate.¹⁸ A characterization of the Clifford group in the nonbinary case can be found in Ref. 19.

Clifford operators are important in the stabilizer formalism because they send any stabilizer state to a stabilizer state. Suppose $|\psi\rangle$ is a stabilizer state with the stabilizer group \mathcal{S} . Furthermore, let L be a Clifford operator. For any $g \in \mathcal{S}$ we have $(LgL^\dagger)L|\psi\rangle = L|\psi\rangle$. On the other hand, LgL^\dagger is in LSL^\dagger which is a subgroup of the Pauli group, since L is a Clifford operator. In fact, LSL^\dagger is a maximal Abelian subgroup of the Pauli group whose corresponding stabilizer state is $L|\psi\rangle$. Therefore, Clifford operators send stabilizer states to stabilizer states.

Based on the characterization of the Clifford group,^{18,19} for any two stabilizer states $|\psi\rangle$ and $|\psi'\rangle$ there is a Clifford operator L such that $L|\psi\rangle = |\psi'\rangle$. However, it does not mean that all the stabilizer states are the same in the point of view of quantum coding theory, since the operator L may completely change the entanglement of a state. But if we assume that $L = L_1 \otimes \dots \otimes L_n$ is a local Clifford operator (L is the tensor product of n one-qubit Clifford operators), then $|\psi\rangle$ and $L|\psi\rangle$ are entangled in the same way. Based on this idea, two stabilizer states are called “local Clifford equivalent” if they are equivalent under the action of the local Clifford group.

For the encoding circuits, we need only two Clifford operators which we describe next. Define the vector

$$|\widehat{r}\rangle = \frac{1}{\sqrt{p}} \sum_{s=0}^{p-1} \omega^{-rs} |s\rangle, \quad (9)$$

for any $r \in \mathbb{F}_p$. $|\widehat{r}\rangle$ is an eigenvector of X , i.e., $X|\widehat{r}\rangle = \omega^r |\widehat{r}\rangle$, and $\{|\widehat{0}\rangle, \dots, |\widehat{p-1}\rangle\}$ is an orthonormal basis. Therefore, the operator

$$H|\widehat{r}\rangle = |r\rangle, \quad (10)$$

which is called the (generalized) Hadamard gate, is unitary. By definition $HXH^\dagger = Z$. It is also easy to see that $HZH^\dagger = X^\dagger$. Hence, H is in the Clifford group. Using the above relations the proof of Lemma 3 is easy.

Lemma 3: Suppose $|\psi\rangle$ is a stabilizer state whose stabilizer group is represented by the $n \times (2n)$ matrix M . Thus, the matrix representation of the stabilizer state $H_i|\psi\rangle$ (Hadamard gate is applied on the i th qubit) is obtained from M by exchanging the i th and $(n+i)$ th columns and then multiplying the i th column by -1 .

The next operator is a two-qubit gate which is called controlled- Z and is defined by

$$C_z|r\rangle|s\rangle = |r\rangle Z^r |s\rangle = Z^s |r\rangle |s\rangle = \omega^{rs} |r\rangle |s\rangle. \quad (11)$$

We have

$$\begin{aligned} C_z(X \otimes I)C_z^\dagger &= X \otimes Z, \\ C_z(I \otimes X)C_z^\dagger &= Z \otimes X, \\ C_z(Z \otimes I)C_z^\dagger &= Z \otimes I, \\ C_z(I \otimes Z)C_z^\dagger &= I \otimes Z, \end{aligned} \quad (12)$$

and thus by definition C_z is in the Clifford group.

Lemma 4: Suppose $|\psi\rangle$ is a stabilizer state whose stabilizer group is represented by the $n \times (2n)$ matrix M . Thus, the matrix representation of the stabilizer state $C_z^{ij}|\psi\rangle$ (the controlled- Z gate is applied on the i th and j th qubits) is obtained from M by adding column i to column $n+j$ and column j to column $n+i$.

D. Graph states

In the following we consider graphs whose edges are labeled by nonzero elements of \mathbb{F}_p . Considering the adjacency matrix of a graph \mathcal{G} , we can represent it by a symmetric matrix over \mathbb{F}_p with zero diagonal. Suppose G is such a matrix of size $n \times n$. Then $M = (I_n | G)$ is a full-rank $n \times (2n)$ matrix, and all of its rows are mutually orthogonal with respect to the symplectic inner product; therefore, M represents a stabilizer group which corresponds to a stabilizer state. Such a stabilizer state is called a *graph state*, which we denote by $|\psi\rangle_{\mathcal{G}}$. It is well-known that any stabilizer state is local Clifford equivalent to a graph state,¹⁷ so to study the properties of stabilizer states it is sufficient to restrict ourselves to graph states.

Graph states can be generated easily using only Hadamard and controlled- Z gates. The corresponding circuit described below is denoted by $G^{(enc)}$.

Lemma 5: The graph state corresponding to the graph with adjacency matrix $G = (g_{ij})$ on n vertices can be generated by the following circuit. Prepare n qubits in the state $|0\rangle$, apply H^\dagger to every one of them, and then for any i, j , apply $C_z^{g_{ij}}$ on qubits i and j . (Note that controlled- Z gates commute.)

Proof: The initial state of the n qubits is $|0\rangle \cdots |0\rangle$, which is a stabilizer state with the stabilizer group $\{Z^{\mathbf{a}} : \mathbf{a} \in \mathbb{F}_p^n\}$. This stabilizer group corresponds to the matrix $M_0 = (\mathbf{0}_n | I_n)$. According to

Lemma 3, after applying H^\dagger gates the matrix M_0 will be changed to $M_1 = (I_n | \mathbf{0}_n)$. Furthermore, by Lemma 4, applying $C_z^{\mathbf{g}_{ij}}$ on qubits i and j corresponds to adding columns i and j multiplied by g_{ij} to columns $n + j$ and $n + i$, respectively. Since the first block of M_1 is identity, this operation is the same as to add g_{ij} to the entries ij and ji of the second block. Therefore, at the end we obtain the matrix $M_2 = (I_n | G)$. \square

E. Measurement on graph states

Suppose we have a graph state $|\psi\rangle_{\mathcal{G}}$ which corresponds to the graph \mathcal{G} with adjacency matrix G , and we measure its (say) last qubit in the standard basis and get $|0\rangle$. We claim that the state after the measurement (without the measured qubit) is also a graph state whose corresponding graph is obtained from \mathcal{G} by removing the last vertex. To see this fact precisely notice that since $(I_n | G)$ represents the stabilizer group of $|\psi\rangle_{\mathcal{G}}$, for any i , we have $X^{\mathbf{e}_i} Z^{\mathbf{g}_i} |\psi\rangle_{\mathcal{G}} = |\psi\rangle_{\mathcal{G}}$, where all coordinates of \mathbf{e}_i are 0 except the i th one which is 1, and \mathbf{g}_i is the i th row of G . Let

$$|\psi\rangle_{\mathcal{G}} = \sum_{r=0}^{p-1} \alpha_r |\phi_r\rangle, \quad (13)$$

and for $1 \leq i \leq n-1$ let \mathbf{g}'_i and \mathbf{e}'_i be the vectors of length $n-1$ obtained from \mathbf{g}_i and \mathbf{e}_i , respectively, by deleting the last coordinate. Thus for $1 \leq i \leq n-1$ we have

$$|\psi\rangle_{\mathcal{G}} = X^{\mathbf{e}_i} Z^{\mathbf{g}_i} |\psi\rangle_{\mathcal{G}} = \sum_{r=0}^{p-1} \alpha_r Z^{\mathbf{g}_{in}} X^{\mathbf{e}'_i} Z^{\mathbf{g}'_i} |\phi_r\rangle |r\rangle = \sum_{r=0}^{p-1} \alpha_r \left(\omega^{r \mathbf{g}_{in}} X^{\mathbf{e}'_i} Z^{\mathbf{g}'_i} |\phi_r\rangle \right) |r\rangle. \quad (14)$$

As a result $X^{\mathbf{e}'_i} Z^{\mathbf{g}'_i} |\phi_0\rangle = |\phi_0\rangle$, which means that $|\phi_0\rangle$ is a stabilizer state with the stabilizer group generated by $X^{\mathbf{e}'_i} Z^{\mathbf{g}'_i}$, $1 \leq i \leq n-1$, and the matrix representation of these generators is $(I_{n-1} | G')$, where G' is the adjacency matrix of the graph obtained from \mathcal{G} by removing its last vertex.

F. CWS codes and graph codes

A CWS code $((n, K, d))_p$ is described by a graph \mathcal{G} with n vertices and edges labeled by \mathbb{F}_p , together with a classical code \mathcal{C} which consists of K vectors in \mathbb{F}_p^n . Such a code is denoted by $\mathcal{Q} = (\mathcal{G}, \mathcal{C})$.⁴⁻⁶

If the classical code \mathcal{C} is linear, then \mathcal{Q} is a graph (stabilizer) code.⁴⁻⁶ The parameters of such a graph code $\mathcal{Q} = (\mathcal{G}, \mathcal{C})$ are $[[n, k, d]]_p$, where the classical code \mathcal{C} consists of $K = p^k$ vectors in \mathbb{F}_p^n that are indexed by the elements of \mathbb{F}_p^k . This $[[n, k, d]]_p$ graph code encodes k qubits into n qubits in the following way. Suppose $|\psi\rangle_{\mathcal{G}}$ is the graph state corresponding to \mathcal{G} . To encode a state of the form $(H^\dagger \otimes \cdots \otimes H^\dagger) |r_1 \dots r_k\rangle$, we first find the classical codeword $\alpha \in \mathcal{C}$, which is indexed by $r_1 \dots r_k$, and then encode $(H^\dagger \otimes \cdots \otimes H^\dagger) |r_1 \dots r_k\rangle$ into $Z^\alpha |\psi\rangle_{\mathcal{G}}$. Since \mathcal{C} is a linear code, it is a linear subspace of \mathbb{F}_p^n . We can then represent \mathcal{C} by k basis vectors $\alpha_1, \dots, \alpha_k$. In this case, the state $(H^\dagger \otimes \cdots \otimes H^\dagger) |r_1 \dots r_k\rangle$ is encoded into $Z^{r_1 \alpha_1 + \cdots + r_k \alpha_k} |\psi\rangle_{\mathcal{G}}$.

Remark 1: $Z^{r_1 \alpha_1 + \cdots + r_k \alpha_k} |\psi\rangle_{\mathcal{G}}$ is usually considered as the encoded state of $|r_1 \dots r_k\rangle$; however, two states $(H^\dagger \otimes \cdots \otimes H^\dagger) |r_1 \dots r_k\rangle$ and $|r_1 \dots r_k\rangle$ are locally equivalent, and then the corresponding codes are equivalent under a local change of basis and have similar properties. Later we will see how these extra Hadamard gates lead us to a simple rule for graph concatenation.

The encoding circuit of an $[[n, k, d]]_p$ graph code is simple.

Procedure 1: (Encoding circuit for a graph code)

1. First generate the graph state $|\psi\rangle_{\mathcal{G}}$ using the circuit described in Lemma 5.
2. For any $1 \leq i \leq k$, apply H^\dagger on q_i , where q_1, \dots, q_k are the qubits that we want to encode.
3. For any $1 \leq j \leq n$, apply $C_z^{\alpha_{ij}}$ on q_i and the j th qubit of $|\psi\rangle_{\mathcal{G}}$, where α_{ij} is the j th coordinate of α_i . Note the entire encoding circuit now involves $n + k$ qubits, where $q_{k+1}, q_{k+2}, \dots, q_{n+k}$ denote the n qubits of $|\psi\rangle_{\mathcal{G}}$.
4. Apply H to q_1, \dots, q_k .

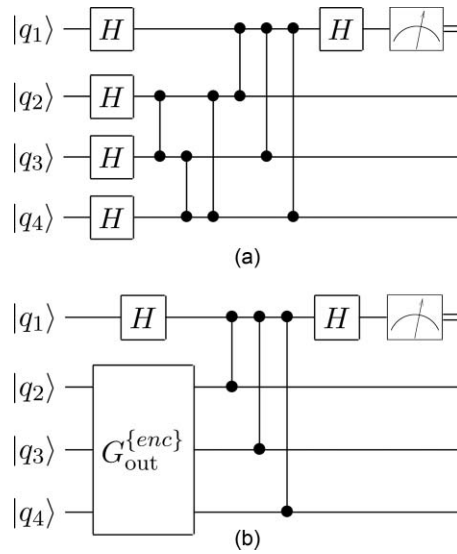


FIG. 1. The encoding circuit of the binary CWS code $\mathcal{Q}_{\text{out}} = (\mathcal{G}_{\text{out}}, \mathcal{C}_{\text{out}})$, where \mathcal{G}_{out} is a triangle and $\mathcal{C}_{\text{out}} = \{000, 111\}$. (Observe that in the binary case, $H^\dagger = H$.) Here q_1 is encoded into qubits q_2, q_3, q_4 . In the left circuit, the three Hadamard gates and the three controlled-Z gates applied on q_2, q_3, q_4 indeed generate the graph state $|\psi\rangle_{\mathcal{G}_{\text{out}}}$. This circuit is simplified in the right figure by replacing those six gates with $G_{\text{out}}^{\text{enc}}$.

5. Measure q_1, \dots, q_k in the computational basis. Throughout the paper we assume that we always get the outcome $|0\rangle$ in the measurements (if not, we just need to perform some local Pauli operations according to the actual measurement outcomes).

Note that the second step of this procedure in which we apply H^\dagger on qubits that are going to be encoded is due to our convention described in Remark 1. For an example of such an encoding circuit, see Fig. 1.

In general, for a graph code $\mathcal{Q} = (\mathcal{G}, \mathcal{C})$, the encoding circuit can be represented graphically, and the corresponding graph is denoted by $\mathcal{G}^{\mathcal{C}}$: consider the graph \mathcal{G} , for any $1 \leq i \leq k$ add a vertex (input vertices), attach it to the vertices of \mathcal{G} (called the output vertices), and label the edge between this vertex and the j th vertex of \mathcal{G} by α_{ij} . For example, Fig. 2(b) gives the graph $\mathcal{G}^{\mathcal{C}}$, where \mathcal{G} is a triangle and $\mathcal{C} = \{000, 111\}$.

Remark 2: The encoding circuit corresponding to the graph code with graphical representation $\mathcal{G}^{\mathcal{C}}$ is related to the circuit that generates the graph state $|\psi\rangle_{\mathcal{G}^{\mathcal{C}}}$ corresponding to the graph $\mathcal{G}^{\mathcal{C}}$. Steps 1, 2, and 3 of Procedure 1 indeed give such a circuit.

Now in order to find the logical X and Z operators, we describe an additive graph code in terms of the stabilizer formalism rather than the CWS formalism. That is, we give the stabilizer group of the logical state $|0 \dots 0\rangle_L$, which uniquely determines this state, and the logical X operators $\bar{X}_1, \dots, \bar{X}_k$. Then the encoded state of $|r_1 \dots r_k\rangle$ is equal to $\bar{X}_1^{r_1} \dots \bar{X}_k^{r_k} |0 \dots 0\rangle_L$.

Observe that

$$|0 \dots 0\rangle = \frac{1}{\sqrt{p^k}} \sum_{r_1, \dots, r_k} (H^\dagger \otimes \dots \otimes H^\dagger) |r_1 \dots r_k\rangle. \quad (15)$$

Then by the definition of graph codes, we have

$$|0 \dots 0\rangle_L = \frac{1}{\sqrt{p^k}} \sum_{r_1, \dots, r_k} Z^{r_1 \alpha_1 + \dots + r_k \alpha_k} |\psi\rangle_{\mathcal{G}}. \quad (16)$$

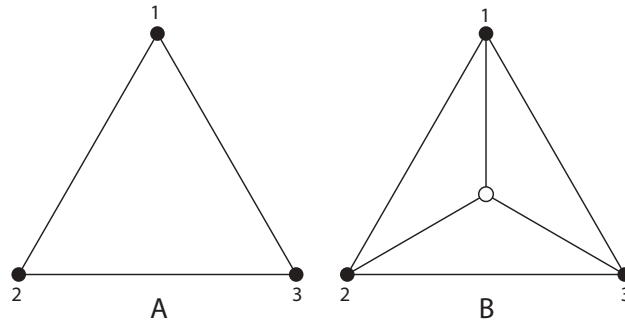


FIG. 2. A linear $[[3, 1, 1]]$ CWS code, where the corresponding graph is a triangle (the left graph) and the classical code is $\{000, 111\}$. Since the classical code is linear (generated by 111), we can show the whole code by one graph. In the right graph, the middle white vertex represents the input vertex which is encoded into vertices 1, 2, 3. This vertex is adjacent to all other vertices because in the generator of the classical code (111), all coordinates are 1.

Therefore, all operators Z^{α_i} are in the stabilizer group of $|0 \dots 0\rangle_L$, and the rows of the matrix

$$\left(\begin{array}{c|c} \mathbf{0} & \alpha_1 \\ \vdots & \vdots \\ \mathbf{0} & \alpha_k \end{array} \right) \quad (17)$$

describe a part of the stabilizer group of $|0 \dots 0\rangle_L$. Since the vectors α_i are linearly independent, without loss of generality (by a change of basis for the classical code and reordering the qubits), we may assume that the first block of the second part of this matrix is I_k . So we assume that the matrix $(I_k \ A)$, where A is of size $k \times (n - k)$ describes a basis for \mathcal{C} .

Assume that

$$G = \begin{pmatrix} G_1 & B \\ B^T & G_2 \end{pmatrix}, \quad (18)$$

where G_1 , G_2 , and B are of size $k \times k$, $(n - k) \times (n - k)$, and $k \times (n - k)$, respectively. Then the stabilizer group of the state $|\psi\rangle_G$ is represented by

$$\left(\begin{array}{c|c} I_k & \mathbf{0} \\ \mathbf{0} & I_{n-k} \end{array} \middle| \begin{array}{cc} G_1 & B \\ B^T & G_2 \end{array} \right). \quad (19)$$

Note that rows of the matrix

$$(-A^T \ I_{n-k} \mid -A^T G_1 + B^T \ -A^T B + G_2) \quad (20)$$

are linear combinations of the rows of Eq. (19), are linearly independent, and are orthogonal to rows of $(I_k \ A)$. Then it is easy to see that the stabilizer group of $|0 \dots 0\rangle_L$ is given by the matrix

$$\left(\begin{array}{c|c} \mathbf{0}_k & \mathbf{0} \\ -A^T & I_{n-k} \end{array} \middle| \begin{array}{cc} I_k & A \\ -A^T G_1 + B^T & -A^T B + G_2 \end{array} \right).$$

On the other hand, for any $1 \leq i, j \leq k$, $(Z^{\alpha_i})(X^{\mathbf{e}_j} Z^{\mathbf{g}_j}) = \omega^{\delta_{ij}} (X^{\mathbf{e}_j} Z^{\mathbf{g}_j})(Z^{\alpha_i})$, where δ_{ij} is the Kronecker delta function (here we use the fact that α_i 's are described by $(I_k \ A)$). Moreover, the other $n - k$ rows of the above matrix are orthogonal to $(\mathbf{e}_j \mid \mathbf{g}_j)$. Therefore, the logical X operators can be described by the matrix

$$(I_k \ \mathbf{0} \mid G_1 \ B).$$

As a summary, the additive graph code \mathcal{Q} is described by the stabilizer group

$$\mathcal{S} = \left(\begin{array}{c|c} \mathbf{0}_k & \mathbf{0} \\ -A^T & I_{n-k} \end{array} \middle| \begin{array}{cc} I_k & A \\ -A^T G_1 + B^T & -A^T B + G_2 \end{array} \right), \quad (21)$$

logical Z operators

$$\mathcal{Z} = (\mathbf{0}_k \mathbf{0} \mid I_k A), \quad (22)$$

and logical X operators

$$\mathcal{X} = (I_k \mathbf{0} \mid G_1 B). \quad (23)$$

G. A simple example

Let us start by taking the outer code to be a binary CWS quantum code $\mathcal{Q}_{\text{out}} = (\mathcal{G}_{\text{out}}, \mathcal{C}_{\text{out}})$, where \mathcal{G}_{out} is a triangle ($n' = 3$) given in Fig. 2(a) and $\mathcal{C}_{\text{out}} = \{000, 111\}$ which is linear. So \mathcal{Q}_{out} is a $[[3, 1, 1]]$ graph code. In this example, the classical encoder is simply given by

$$0 \rightarrow 000, 1 \rightarrow 111; \quad (24)$$

however, in general, when \mathcal{C}_{out} is not linear, the encoding circuit of \mathcal{Q}_{out} is started by a *classical encoder*. The encoding circuit of this code is given in Fig. 1.

Now we consider the inner code $\mathcal{Q}_{\text{in}} = (\mathcal{G}_{\text{in}}, \mathcal{C}_{\text{in}})$. Due to our restriction for Eq. (2), \mathcal{C}_{in} must be linear. So \mathcal{Q}_{in} is a graph code and has a graph representation $\mathcal{G}_{\text{in}}^{\mathcal{C}_{\text{in}}}$. Let \mathcal{Q}_{in} be a $[[2, 1, 1]]$ stabilizer code, which is represented by the graph of Fig. 3(a) on the vertices 1, 4, 5: the subgraph of the vertices 4, 5 represents \mathcal{G}_{in} and 1 is the input qubit describing the classical encoder of \mathcal{C}_{in} ; hence, $\mathcal{C}_{\text{in}} = \{00, 11\}$.

The encoding circuit of the concatenated code $\mathcal{Q}_c = \mathcal{Q}_{\text{in}} \sqsubset \mathcal{Q}_{\text{out}}$ is given by Fig. 4, where $G_{\text{in}}^{\text{enc}}$ denotes the graph encoder for the graph of the inner code \mathcal{G}_{in} . In this circuit, we first encode the qubit q_0 into three qubits q_2, q_3, q_4 using the encoding circuit of \mathcal{Q}_{out} and then encode each one of these three qubits according to \mathcal{Q}_{in} . Note that the three Hadamard gates in the middle of the circuit applied to q_2, q_3, q_4 are again because of the convention of adding a Hadamard gate before performing the classical encoder (see Remark 1).

This circuit can also be shown by a graph as in Fig. 3(b). Here, in the outer code, the middle white vertex is encoded into vertices 1, 2, 3. Then each of these vertices is encoded using the inner code: vertex 1 into vertices 4, 5; vertex 2 into vertices 6, 7; and vertex 3 into vertices 8, 9. We call this graphical representation of the concatenated code \mathcal{Q}_c with a linear \mathcal{C}_{out} the encoding graph of \mathcal{Q}_c and denote it by $\mathcal{G}_{\mathcal{Q}_c}^{\mathcal{C}_{\text{out}}^{\text{enc}}}$. We have three types of vertices in $\mathcal{G}_{\mathcal{Q}_c}^{\mathcal{C}_{\text{out}}^{\text{enc}}}$: the input vertices (the middle white vertex in our example); auxiliary vertices which are in the subgraph \mathcal{G}_{out} (vertices 1, 2, 3); and output vertices which are in the copies of \mathcal{G}_{in} (vertices 4, 5, 6, 7, 8, 9). In general, if \mathcal{C}_{out} is nonlinear, similarly we can have an encoding graph of \mathcal{Q}_c and denote it by $\mathcal{G}_{\mathcal{Q}_c}^{\text{enc}}$, which in our example is the subgraph of Fig. 3(b) without the middle white vertex.

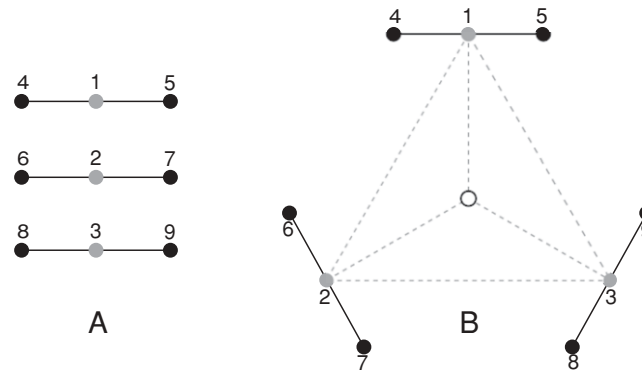


FIG. 3. The right graph shows $\mathcal{G}_{\mathcal{Q}_c}^{\mathcal{C}_{\text{out}}^{\text{enc}}}$ when the outer code is given in Fig. 2 (here depicted by gray dots) and the inner code is given by the three vertices 1, 4, 5 on the left graph: 1 is the input vertex and 4, 5 are the output vertices.

TABLE I. Notations.

\mathcal{C}	The classical code
C	The matrix whose rows form a basis for \mathcal{C} , if \mathcal{C} is linear (the generator matrix)
\mathcal{G}	The graph corresponding to the graph state $ \psi\rangle_{\mathcal{G}}$
G	The adjacency matrix of the graph \mathcal{G} ($G = (g_{ij})$)
$\mathcal{G}^{\mathcal{C}}$	The graph representing the graph code \mathcal{Q} , if \mathcal{C} is linear
$G^{\{enc\}}$	The encoding circuit of the graph \mathcal{G} given by Lemma 5
$\mathcal{G}_{\mathcal{Q}_c}^{\{enc\}}$	The encoding graph of the concatenated code \mathcal{Q}_c
$\mathcal{G}_{\mathcal{Q}_c}^{\mathcal{C}_{out}\{enc\}}$	The encoding graph of the concatenated code \mathcal{Q}_c , if \mathcal{C}_{out} is linear

H. Summary of notations

Before going into the detailed proof of the main result, we summarize our notation. Let $\mathcal{Q} = (\mathcal{G}, \mathcal{C})$ be a CWS code. If \mathcal{C} is linear, then \mathcal{Q} is a graph code, where the code has a graphical representation denoted by $\mathcal{G}^{\mathcal{C}}$. The concatenation of two CWS quantum codes $\mathcal{Q}_{in} = (\mathcal{G}_{in}, \mathcal{C}_{in})$ and $\mathcal{Q}_{out} = (\mathcal{G}_{out}, \mathcal{C}_{out})$ is denoted by $\mathcal{Q}_c = \mathcal{Q}_{in} \sqsubset \mathcal{Q}_{out}$. See Table I for the rest of notations.

III. CONCATENATION OF GRAPH CODES

In this section, we prove our main result in a simple case, where the inner code encodes only a single qupit and the outer code is a graph code. In this situation, we can algebraically obtain the graph and the classical code of the concatenated code using the stabilizer formalism. Although we will prove our main result in the general case in Sec. IV, we believe that the proof given in this section is easily accessible to those who are familiar with the stabilizer formalism.

Suppose the inner code $\mathcal{Q}_{in} = (\mathcal{G}_{in}, \mathcal{C}_{in})$ encodes only a single qupit, i.e., \mathcal{Q}_{in} is an $[[n, 1, d]]_p$ code. Then from the discussion in Sec. II F, it follows that

$$G_{in} = \begin{pmatrix} 0 & \mathbf{y} \\ \mathbf{y}^T & H' \end{pmatrix} \quad (25)$$

and

$$\mathcal{C}_{in} = \{\mathbf{0}_{1 \times n}, (1 \ \mathbf{b})\}, \quad (26)$$

i.e.,

$$C_{in} = (1 \ \mathbf{b}), \quad (27)$$

where both \mathbf{y} and \mathbf{b} are vectors of length $n - 1$. (Note that \mathcal{Q}_{in} encodes one qupit; thus, C_{in} is one-dimensional.)

Since \mathcal{Q}_{in} encodes one qupit, the corresponding outer code $\mathcal{Q}_{out} = (\mathcal{G}_{out}, \mathcal{C}_{out})$ is an $((n', K', d'))_p$ code. In this section, we assume that \mathcal{C}_{out} is linear, so \mathcal{Q}_{out} is a graph code with parameters $[[n', k', d']]_p$, where $K' = p^{k'}$. Then from the discussion in Sec. II F, we have

$$G_{out} = \begin{pmatrix} G_1 & B \\ B^T & G_2 \end{pmatrix}, \quad (28)$$

and the rows of

$$C_{out} = (I_{k'} \ A) \quad (29)$$

form a basis for \mathcal{C}_{out} .

Thus by Eqs. (21)–(23), the stabilizer group of \mathcal{Q}_{in} is

$$\mathcal{S}_{in} = \left(\begin{array}{cc|c} 0 & \mathbf{0} & 1 \\ -\mathbf{b}^T & I_{n-1} & \mathbf{y}^T - \mathbf{b}^T \mathbf{y} + H' \end{array} \right), \quad (30)$$

its logical operator Z is given by

$$\mathcal{Z}_{in} = (0 \ \mathbf{0} \ | \ 1 \ \mathbf{b}), \quad (31)$$

and its logical operator X is given by

$$\mathcal{X}_{\text{in}} = (1 \ 0 \mid 0 \ \mathbf{y}). \quad (32)$$

The concatenated code $\mathcal{Q}_c = \mathcal{Q}_{\text{in}} \sqsubset \mathcal{Q}_{\text{out}}$ is a quantum code which encodes k' qupits into nn' qupits as follows: it first encodes k' qupits into n' qupits using \mathcal{Q}_{out} and then encodes any of the n' qupits into n qupits based on \mathcal{Q}_{in} .

The main result of this section is given by Theorem 1, which states that the concatenated code $\mathcal{Q}_c = \mathcal{Q}_{\text{in}} \sqsubset \mathcal{Q}_{\text{out}}$ is also a graph code. The corresponding adjacency matrix of the graph and the generator matrix of the classical code can be computed directly from the adjacency matrices and the generator matrices of the inner and outer codes.

Theorem 1: Suppose $\mathcal{Q}_{\text{out}} = (\mathcal{G}_{\text{out}}, \mathcal{C}_{\text{out}})$ and $\mathcal{Q}_{\text{in}} = (\mathcal{G}_{\text{in}}, \mathcal{C}_{\text{in}})$ are $[[n', k', d']]_p$ and $[[n, k, d]]_p$ graph codes, respectively, (where $k = 1$) as described by Eqs. (25)–(29). Then the concatenated code $\mathcal{Q}_c = \mathcal{Q}_{\text{in}} \sqsubset \mathcal{Q}_{\text{out}} = (\mathcal{G}_c, \mathcal{C}_c)$ is a graph code described by the graph \mathcal{G}_c with adjacency matrix

$$G_c = G_{\text{in}} \otimes I_{n'} + \begin{pmatrix} 1 \\ \mathbf{b}^T \end{pmatrix} (1 \ \mathbf{b}) \otimes G_{\text{out}}, \quad (33)$$

and the classical code with generator matrix

$$C_c = (1 \ \mathbf{b}) \otimes (I_{k'} \ A), \quad (34)$$

i.e., the classical code is obtained by concatenation as well,

$$\mathcal{C}_c = \mathcal{C}_{\text{in}} \sqsubset \mathcal{C}_{\text{out}}. \quad (35)$$

Proof: Let us first show that a basis of \mathcal{C}_c is described by Eq. (34). To find \mathcal{Z}_c , the logical Z operators of \mathcal{Q}_c , we should first consider the logical Z operators of \mathcal{Q}_{out} and then replace any Pauli matrix X and Z of those operators with the logical X and Z operators of \mathcal{Q}_{in} . For example, if the logical Z operator acting on the first encoded qupit in \mathcal{Q}_{out} is $Z^{(1,1,0,\dots,0)}$, the logical Z operator acting on the first qupit in \mathcal{Q}_c is $Z^{((1,\mathbf{b}), (1,\mathbf{b}), \mathbf{0}, \dots, \mathbf{0})}$ because by Eq. (31) the logical Z operator of \mathcal{Q}_{in} is $Z^{(1,\mathbf{b})}$. Therefore, by changing the order of the qupits we can represent this Pauli matrix by the vector $(\mathbf{0} \mid (1 \ \mathbf{b}) \otimes (1, 1, 0, \dots, 0))$, where the zero before the vertical line is actually a zero vector. Now since the logical Z operators of \mathcal{Q}_{out} are represented by rows of Eq. (22), we have

$$\mathcal{Z}_c = (\mathbf{0} \mid (1 \ \mathbf{b}) \otimes (I_{k'} \ A)). \quad (36)$$

Equivalently, $(1 \ \mathbf{b}) \otimes (I_{k'} \ A)$ is a basis for the linear code \mathcal{C}_c .

Analogously, we compute for the logical X operators of \mathcal{Q}_c

$$\mathcal{X}_c = ((1 \ \mathbf{0}) \otimes (I_{k'} \ \mathbf{0}) \mid (0 \ \mathbf{y}) \otimes (I_{k'} \ \mathbf{0}) + (1 \ \mathbf{b}) \otimes (G_1 \ B)) \quad (37)$$

$$= (I_k \ \mathbf{0} \mid G_1 \ B \ \mathbf{b} \otimes (G_1 \ B) + \mathbf{y} \otimes (I_{k'} \ \mathbf{0})). \quad (38)$$

It remains to compute the stabilizer group. The first n' rows of \mathcal{S}_c are obtained from rows of \mathcal{S}_{out} [Eq. (21)] by replacing any X and Z with the logical X and Z of the inner code. For the next $(n-1)n'$ rows, note that g_2, \dots, g_n which are the Pauli matrices corresponding to the rows 2, \dots , n of \mathcal{S}_{in} commute with the logical X and Z of the inner code. In fact, they are in the stabilizer group of the code space (spanned by the states $|0\rangle, \dots, |p-1\rangle$). Now since we replace any of the n' qupits of \mathcal{Q}_{out} with a state in the code space of \mathcal{Q}_{in} , each block of n qupits in \mathcal{Q}_c should be stabilized by g_2, \dots, g_n . As a result, $\mathcal{S}_c = (M \mid N)$, where

$$M = \begin{pmatrix} (1 \ \mathbf{0}) \otimes \begin{pmatrix} \mathbf{0}_k & \mathbf{0} \\ -A^T & I_{n'-k'} \end{pmatrix} \\ (-\mathbf{b}^T \otimes I_{n-1}) \otimes I_{n'} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_k & \mathbf{0} & \mathbf{0} \\ -A^T & I_{n'-k'} & \mathbf{0} \\ -\mathbf{b}^T \otimes \begin{pmatrix} I_{k'} \\ \mathbf{0} \end{pmatrix} & -\mathbf{b}^T \otimes \begin{pmatrix} \mathbf{0} \\ I_{n'-k'} \end{pmatrix} & I_{(n-1)n'} \end{pmatrix} \quad (39)$$

and

$$N = \left(\begin{array}{c} (0 \ \mathbf{y}) \otimes \left(\begin{array}{cc} \mathbf{0}_k & \mathbf{0} \\ -A^T & I_{n'-k'} \end{array} \right) + (1 \ \mathbf{b}) \otimes \left(\begin{array}{c} I_{k'} \\ -A^T G_1 + B^T - A^T B + G_2 \end{array} \right) \\ (\mathbf{y}^T \ -\mathbf{b}^T \mathbf{y} + H') \otimes I_{n'} \end{array} \right). \quad (40)$$

Now to complete the proof of Theorem 1 it is sufficient to show that the stabilizer group and the logical X and Z operators of the graph code described by Eqs. (33) and (34) are given by Eqs. (36)–(40). We compute these matrices using the construction given by Eqs. (21)–(23).

First of all, the classical part of the code is given by $(I_{k'} \ B)$, where $B = (A \ \mathbf{b} \otimes (I_{k'} \ A))$; therefore, the logical Z operators of the code are the same as Eq. (36).

The block from of matrix G_c of Eq. (33) is given by

$$G_c = \left(\begin{array}{cc} K_1 & W \\ W^T & K_2 \end{array} \right), \quad (41)$$

where $K_1 = G_1$, $W = (B \ \mathbf{y} \otimes (I_{k'} \ \mathbf{0}) + \mathbf{b} \otimes (G_1 \ B))$ and

$$K_2 = \left(\begin{array}{cc} G_2 & \mathbf{y} \otimes (\mathbf{0} \ I_{n'-k'}) + \mathbf{b} \otimes (B^T \ G_2) \\ \mathbf{y}^T \otimes \left(\begin{array}{c} \mathbf{0} \\ I_{n'-k'} \end{array} \right) + \mathbf{b}^T \otimes \left(\begin{array}{c} B \\ G_2 \end{array} \right) & H' \otimes I_{n'} + \mathbf{b}^T \mathbf{b} \otimes G_2 \end{array} \right). \quad (42)$$

Hence, the logical X operator of the graph code is

$$(I_{k'} \ \mathbf{0} \mid K_1 \ W) = (I_{k'} \ \mathbf{0} \mid K_1 \ B \ \mathbf{y} \otimes (I_{k'} \ \mathbf{0}) + \mathbf{b} \otimes (G_1 \ B)), \quad (43)$$

which is the same as Eq. (37).

The stabilizer group of the graph code is given by

$$\left(\begin{array}{cc|c} \mathbf{0}_{k'} & \mathbf{0} & I_{k'} \\ -B^T & I_{nn'-k'} & -B^T K_1 + W^T - B^T W + K_2 \end{array} \right). \quad (44)$$

By Lemma 2 this matrix describes the same group as $\mathcal{S}_c = (M \mid N)$ because we have

$$\mathcal{S}_c = \left(\begin{array}{ccc} I_{k'} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{n'-k'} & \mathbf{0} \\ \mathbf{0} & -\mathbf{b}^T \otimes \left(\begin{array}{c} \mathbf{0} \\ I_{n'-k'} \end{array} \right) & I_{(n-1)n'} \end{array} \right) \left(\begin{array}{cc|c} \mathbf{0}_{k'} & \mathbf{0} & I_{k'} \\ -B^T & I_{nn'-k'} & -B^T K_1 + W^T - B^T W + K_2 \end{array} \right). \quad (45)$$

□

Note that from Eq. (33), the adjacency matrix G_c does not depend on the classical code of the outer code (\mathcal{C}_{out}), which indicates that Theorem 1 could also be true even if \mathcal{C}_{out} is nonlinear (in this case Eq. (34) does no longer apply, but Eq. (35) may still hold). As the stabilizer formalism can no longer be used to handle this case, we need an alternative proof technique which will be presented in Sec. IV. This new technique is based on analyzing the encoding circuit of the concatenated code. It can easily be extended to more general cases, such as nonlinear outer codes, $k > 1$, and even to generalized concatenated quantum codes.

In the example of Sec. II G, we have $A = (1 \ 1)$, $(1 \ \mathbf{b}) = (1 \ 1)$, $G_{\text{in}} = 0$, and

$$G_{\text{out}} = \left(\begin{array}{ccc} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \right).$$

Therefore, $C_c = (1 \ 1 \ 1 \ 1 \ 1 \ 1)$ and

$$G_c = \left(\begin{array}{cccccc} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{array} \right). \quad (46)$$

IV. GRAPH CONCATENATION BY GENERALIZED LOCAL COMPLEMENTATION

In this section we prove our main result based on analyzing the encoding circuits of the concatenated code. We start with an alternative proof for Theorem 1 in Sec. IV A for the simple case that the inner code encodes only a single qubit and the outer code is a graph code. This proof is based on the rule of generalized local complementation. Then in Sec. IV B, we show that the rule of generalized local complementation can be directly applied to the case that the outer code is a general CWS code, which is beyond the result of Theorem 1. In Sec. IV C, we discuss the case where the inner code encodes more than one qubit (i.e., $k > 1$); we show that the rule of generalized local complementation given in Sec. IV A also applies directly to this case and hence completes the proof of the main result.

A. Alternative proof for Theorem 1

Recall our main goal: suppose we have two graph codes $\mathcal{Q}_{\text{out}} = (\mathcal{G}_{\text{out}}, \mathcal{C}_{\text{out}})$ and $\mathcal{Q}_{\text{in}} = (\mathcal{Q}_{\text{in}}, \mathcal{C}_{\text{in}})$ given by Eqs. (25)–(29), where \mathcal{Q}_{in} encodes a single qubit. Let $\mathcal{Q}_c = \mathcal{Q}_{\text{in}} \sqsubset \mathcal{Q}_{\text{out}}$ denote the concatenation of the inner code \mathcal{Q}_{in} and the outer code \mathcal{Q}_{out} . We would like to show that $\mathcal{Q}_c = (\mathcal{Q}_c, \mathcal{C}_c)$, where \mathcal{Q}_c and \mathcal{C}_c are given in Eqs. (33) and (34), respectively.

In Sec. IV A 1, we first specify the encoding circuit of the concatenated code \mathcal{Q}_c and then we give the graphical interpretation of this circuit and define the encoding graph $\mathcal{G}_{\mathcal{Q}_c}^{\mathcal{C}_{\text{out}}\{\text{enc}\}}$ of the concatenated code \mathcal{Q}_c . Then in Sec. IV A 2, we define the rule of generalized local complementation on a graph; we show how the encoding circuit of the concatenated code \mathcal{Q}_c can be interpreted as generalized local complementation on the encoding graph and how we can obtain the graph code $\mathcal{G}_c^{\mathcal{C}_c}$ from the encoding graph $\mathcal{G}_{\mathcal{Q}_c}^{\mathcal{C}_{\text{out}}\{\text{enc}\}}$; finally, we show that \mathcal{Q}_c and \mathcal{C}_c are exactly those given in Eqs. (33) and (34), thereby completing the proof.

1. Encoding circuit and encoding graph for the concatenated code

We have already discussed the encoding circuit of a concatenated code in the example of Sec. II G. Here we state it more formally.

Procedure 2: (Encoding circuit for \mathcal{Q}_c with a graph outer code and an inner code encoding a single qubit)

1. Apply the encoding circuit of \mathcal{Q}_{out} that encodes k' qubits into n' qubits which we call $q_1, \dots, q_{n'+k'}$, as given by Procedure 1.
2. Apply n' copies of the circuit that gives the graph state corresponding to \mathcal{G}_{in} .
3. Apply H^\dagger on all qubits $q_{k'+1}, \dots, q_{n'+k'}$.
4. Apply the corresponding controlled-Z operators between these qubits and the graph states of \mathcal{G}_{in} .
5. Apply H on $q_1, \dots, q_{n'+k'}$.
6. Measure $q_1, \dots, q_{n'+k'}$ in the computational basis.

As discussed in Sec. II G, Procedure 2 can be represented by a graph which is denoted by $\mathcal{G}_{\mathcal{Q}_c}^{\mathcal{C}_{\text{out}}\{\text{enc}\}}$. This graph is constructed as follows: step 1 corresponds to the graph \mathcal{G}_{out} ; step 2 corresponds to adding a copy of the graph \mathcal{G}_{in} for each vertex q_i of the graph \mathcal{G}_{out} and then we have a graph on $n' + k' + nn'$ vertices; steps 3, 4, and 5 encode the n' qubits of the outer code into n' copies of the inner code. Hence we just add edges and labels according to controlled-Z gates that are applied between these n' qubits and the graph states of \mathcal{G}_{in} . The n' vertices $q_1, \dots, q_{n'}$ which will be measured at the end are called the *auxiliary vertices* of $\mathcal{G}_{\mathcal{Q}_c}^{\mathcal{C}_{\text{out}}\{\text{enc}\}}$. Furthermore, the nn' vertices corresponding to n' copies of \mathcal{G}_{in} are called the *output vertices*.

For an example of the encoding graph $\mathcal{G}_{\mathcal{Q}_c}^{\mathcal{C}_{\text{out}}\{\text{enc}\}}$, see Fig. 3(b). The corresponding encoding circuit is given by Fig. 4. The auxiliary vertices in this example are 1, 2, 3, and the output vertices are 4, 5, 6, 7, 8, 9.

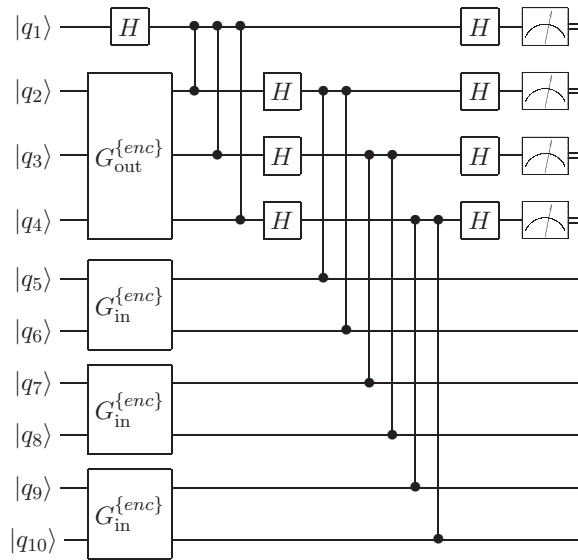


FIG. 4. The encoding circuit of the concatenated code $\mathcal{Q}_c = \mathcal{Q}_{in} \sqcup \mathcal{Q}_{out}$, where \mathcal{Q}_{in} and \mathcal{Q}_{out} are described by Fig. 3.

2. Graph concatenation via generalized local complementation

We now give a graphical interpretation of steps 3, 4, and 5 given in Procedure 2. Note that for $1 \leq i \leq n'$, we apply H^\dagger to qubit q_i , then the corresponding controlled-Z operations between q_i and the i th copy of the graph state \mathcal{G}_{in} , and finally H on q_i . We show that each of these n' steps is equivalent to a generalized local complementation on the graph $\mathcal{G}_{\mathcal{Q}_c}^{enc}$.

Definition 1: (Generalized local complementation) Suppose $F = (f_{ij})$ is the adjacency matrix of a graph \mathcal{F} , i is a vertex of \mathcal{F} , and \mathbf{f}_i is the i th row of F . Furthermore, let \mathbf{v} be a row vector whose coordinates are indexed by the vertices of \mathcal{F} such that \mathbf{v} is zero on i and its neighbors, i.e., $v_j = 0$ if $j = i$ or $f_{ij} \neq 0$. Then the generalized local complementation at (i, \mathbf{v}) is the operation which sends F to $F + \mathbf{v}^T \mathbf{f}_i + \mathbf{f}_i^T \mathbf{v}$.

Note that since \mathbf{v} is zero on the neighbors of i , for any j and k , either $(\mathbf{v}^T \mathbf{f}_i)_{jk}$ or $(\mathbf{f}_i^T \mathbf{v})_{jk}$ is equal to zero.

To get an idea on why we call this operation the generalized local complementation, let us consider the binary case.²⁰ In this special case, \mathbf{v} corresponds to a subset of vertices (j belongs to this set iff $v_j = 1$). Then this operation is the same as to replace the bipartite graph induced on the neighbors of i and the vertices in \mathbf{v} with its complement. For an example, see Fig. 5.

Theorem 2: (Encoding circuit interpreted as generalized local complementation) Consider a circuit which corresponds to a graph \mathcal{F} with the adjacency matrix F . Let i be a vertex of \mathcal{F} (or equivalently a qubit in the circuit), and let \mathbf{v} be a vector which is zero on i and its neighbors. Suppose we change the circuit by applying H^\dagger on the i th qubit, $C_z^{v_j}$ (v_j is the j th coordinate of \mathbf{v}) on qubits i and j , for any j , and then H on the i th qubit. Then the resulting circuit is equivalent to the graph \mathcal{F} after the generalized local complementation at (i, \mathbf{v}) .

Proof: For simplicity assume $i = 1$, and let $\mathbf{f}_1 = (0 \ \mathbf{s})$, where \mathbf{f}_1 is the first row of F . Furthermore, let $\mathbf{v} = (0 \ \mathbf{v}')$ and \mathcal{F}' be the graph obtaining from \mathcal{F} by removing its first vertex (and F' its adjacency matrix). Then the stabilizer group corresponding to the circuit is represented by

$$(I | F) = \left(\begin{array}{c|c} 1 & \mathbf{0} \\ \mathbf{0} & I \end{array} \middle| \begin{array}{c|c} 0 & \mathbf{s} \\ \mathbf{s}^T & F' \end{array} \right). \quad (47)$$

Then based on the translation of the action of the Hadamard gate and controlled-Z gate on the stabilizer group (Lemmas 3 and 4), we can compute the stabilizer group after applying those gates

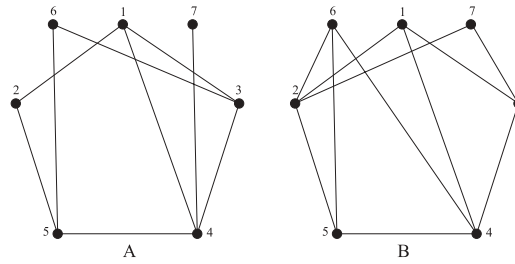


FIG. 5. The generalized local complementation of the graph A at vertex 1 with respect to $S = \{6, 7\}$ results in the graph B. Here neighbors of 1 are $\{2, 3, 4\}$, so this operation replaces the induced bipartite graph on $\{2, 3, 4\} \cup S$ with its complement.

as follows:

$$\left(\begin{array}{c|c} 1 & \mathbf{0} \\ \mathbf{0} & I \end{array} \middle| \begin{array}{c|c} \mathbf{0} & \mathbf{s} \\ \mathbf{s}^T & F' \end{array} \right) \xrightarrow{H^\dagger} \left(\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \mathbf{s}^T & I \end{array} \middle| \begin{array}{c|c} -1 & \mathbf{s} \\ \mathbf{0} & F' \end{array} \right) \xrightarrow{C_z^{1,\mathbf{v}'}} \left(\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \mathbf{s}^T & I \end{array} \middle| \begin{array}{c|c} -1 & \mathbf{s} \\ \mathbf{v}'^T & F' + \mathbf{s}^T \mathbf{v}' \end{array} \right) \quad (48)$$

$$\xrightarrow{H} \left(\begin{array}{c|c} 1 & \mathbf{0} \\ -\mathbf{v}'^T & I \end{array} \middle| \begin{array}{c|c} \mathbf{0} & \mathbf{s} \\ \mathbf{s}^T & F' + \mathbf{s}^T \mathbf{v}' \end{array} \right). \quad (49)$$

Now to relate this stabilizer group to a graph, we change the set of generators by multiplying the above matrix by

$$\left(\begin{array}{c|c} 1 & \mathbf{0} \\ \mathbf{v}'^T & I \end{array} \right), \quad (50)$$

which gives

$$\left(\begin{array}{c|c} 1 & \mathbf{0} \\ \mathbf{0} & I \end{array} \middle| \begin{array}{c|c} \mathbf{0} & \mathbf{s} \\ \mathbf{s}^T & F' + \mathbf{s}^T \mathbf{v}' + \mathbf{v}'^T \mathbf{s} \end{array} \right). \quad (51)$$

Hence, the adjacency matrix F of the graph is changed to $F + \mathbf{v}'^T \mathbf{f}_i + \mathbf{f}_i^T \mathbf{v}'$. \square

Based on this theorem, we can find the graph corresponding to the concatenated code \mathcal{Q}_c .

Procedure 3: (Graph concatenation via generalized local complementation)

1. Given the graph $\mathcal{G}_{\mathcal{Q}_c}^{\{enc\}}$ (or $\mathcal{G}_{\mathcal{Q}_c}^{C_{out}\{enc\}}$), for each auxiliary vertex i , let S_i be the i th row of the adjacency matrix of $\mathcal{G}_{\mathcal{Q}_c}^{\{enc\}}$ in which the label of all auxiliary vertices is changed to zero. (So the vectors S_i have no nonzero coordinate corresponding to auxiliary vertices.) In the binary case, for example, S_i corresponds to the set of all output vertices which are adjacent to i .
2. For each auxiliary vertex i , delete all the edges which connect i to the output vertices. (Change the label of those edges to zero.)
3. For each auxiliary vertex i , perform generalized local complementation at (i, S_i) .
4. Remove all the auxiliary vertices.

Remark 3: Note that two generalized local operations at (i, S_i) and (j, S_j) do not commute in general. However, in the above procedure, the order in which we apply those generalized local complementations does not matter because, as we will show in Corollary 1, the whole procedure on all auxiliary vertices gives the same graph.

See Fig. 6 for the illustration of Procedure 3 for the example of Sec. II G.

Corollary 1: $\mathcal{Q}_c = (\mathcal{G}_c, C_c)$ is a CWS code, and the graph $\mathcal{G}_c^{C_c}$ can be obtained from the encoding graph $\mathcal{G}_{\mathcal{Q}_c}^{C_{out}\{enc\}}$ via Procedure 3.

Proof: We can split parts 3,4, and 5 of Procedure 2 into n' steps, one step for each auxiliary qubit q_i . According to Theorem 2, these n' steps correspond to part 3 of Procedure 3. Measurement of qubits q_i corresponds to removing the auxiliary vertices. \square

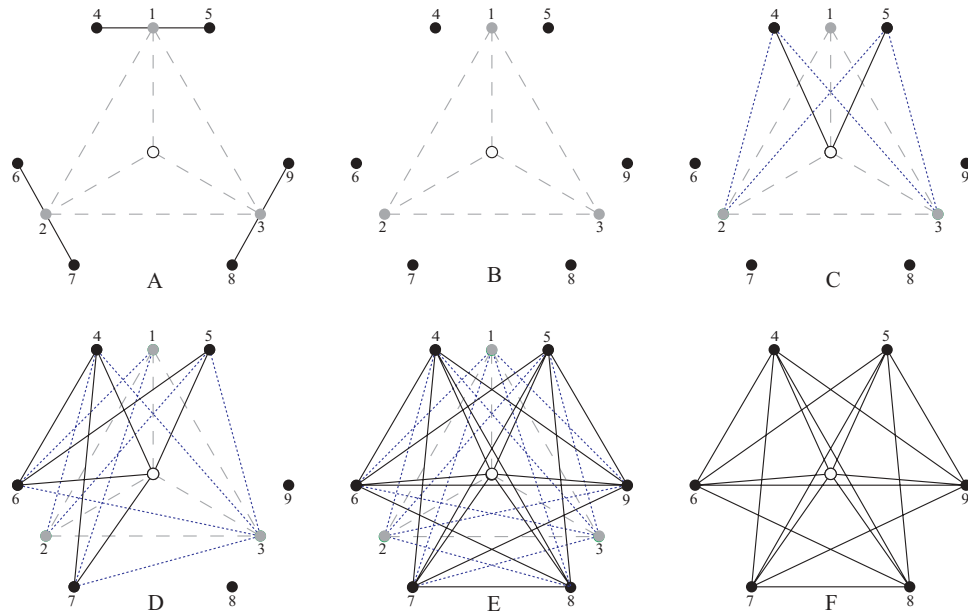


FIG. 6. (Color online) This figure demonstrates Procedure 3 for the inner and outer codes of Sec. II G. The graph $\mathcal{G}_{\mathcal{Q}_c}^{C_{out}\{enc\}}$ is given in part A. The auxiliary vertices are $\{1, 2, 3\}$ and the output vertices are $\{4, 5, 6, 7, 8, 9\}$. Then we have $S_1 = \{4, 5\}$, $S_2 = \{6, 7\}$, and $S_3 = \{8, 9\}$. Removing all edges which connect i to S_i (for $i = 1, 2, 3$) gives the graph shown in part B. Parts C, D, and E are the resulting graphs after applying the generalized local complementation at $(1, S_1)$, $(2, S_2)$, and $(3, S_3)$, respectively. Here edges connecting an output vertex to an auxiliary vertex are depicted by (blue) short-dashed lines. The final graph after removing the auxiliary vertices is shown in part F. Observe that the adjacency matrix of the induced subgraph on the output vertices is equal to Eq. (46).

Note that Corollary 1 proves our main result in the case that the inner code encodes a single qubit and the outer code is linear. Additionally, note that the resulting graph of Corollary 1 is consistent with the one given by Theorem 1 because they both compute the same graph. Here we provide a more direct proof for this fact.

Theorem 3: *The adjacency matrix of the graph $\mathcal{G}_c^{C_c}$ provided by Corollary 1 is given by Eqs. (33) and (34).*

Proof: We briefly describe a proof only for the binary case and for the validity of Eq. (33). This proof can simply be captured for the more general setting.

Based on Procedure 3, the graph on which we apply the generalized local complementation operators has the following subgraphs: \mathcal{G}_{out} with auxiliary vertices $\{1, \dots, n'\}$ and a copy of \mathcal{G}_{in} with vertex set V_i for each auxiliary vertex $1 \leq i \leq n'$. Then for each $1 \leq i \leq n'$, we apply the generalized local complementation on i with respect to $S_i \subseteq V_i$ which is defined based on the classical inner code.

Fact 1: *Eq. (33) describes the unique graph on the vertex set $\bigcup_i V_i$ with the following structure:*

1. *The induced subgraph on V_i , for every i , is isomorphic to \mathcal{G}_{in} .*
2. *For every $i \neq j$, there is no edge between vertices in V_i and $V_j \setminus S_j$.*
3. *For every $i \neq j$, there is an edge between vertices $v \in S_i$ and $w \in S_j$ iff i and j are connected in \mathcal{G}_{out} .*

Clearly the graph with these properties is unique, and moreover Eq. (33) represents this unique graph.

Based on this fact, we will show that the graph resulting from Procedure 3 has the above structure.

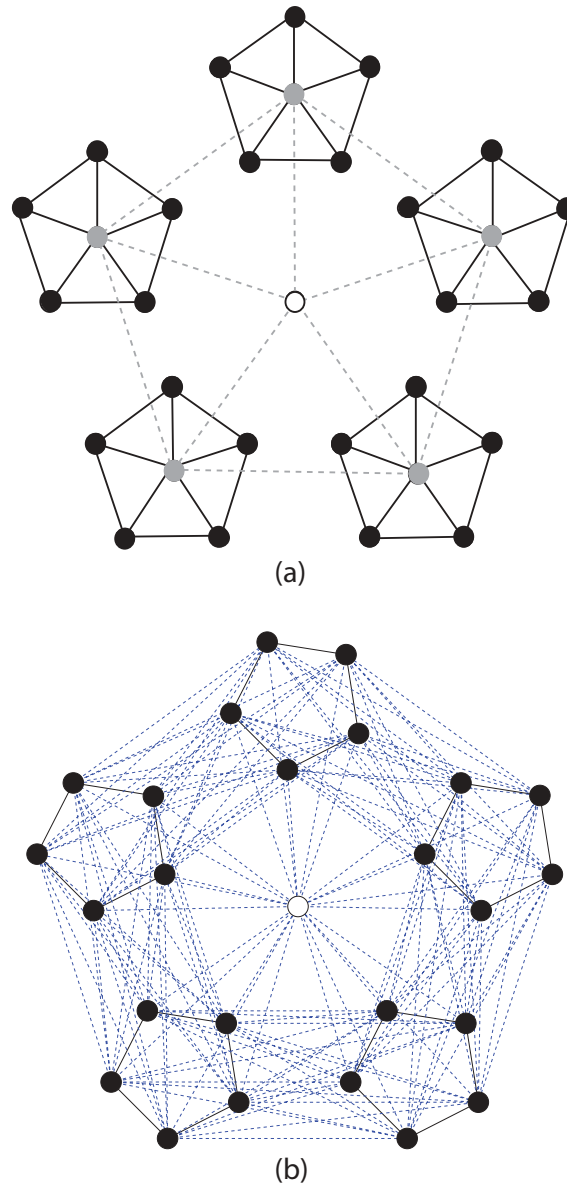


FIG. 7. (Color online) Self-concatenation of the code $[[5, 1, 3]]$ yielding a code $[[25, 1, 9]]$.

Fact 2: During Procedure 3 the changes on the subgraph induced on $\bigcup V_i$ happen only among vertices $v \in S_i$ and $w \in S_j$ for $i \neq j$. As a result, the final graph of Procedure 3 satisfies properties 1 and 2 of Fact 1.

This is simply because in the generalized local complementations we never touch vertices in $V_i \setminus S_i$. Furthermore, in each step, S_i is disjoint from $N(i)$ (the neighbors of vertex i), so there is no change in the subgraph induced on the vertex set S_i .

Fact 3: In Procedure 3, suppose we have applied the generalized local complementation on vertices $1, 2, \dots, l$, for some $1 \leq l \leq n'$. Then for any choice of $v_i \in S_i$, for $1 \leq i \leq l$, the induced subgraph on vertices $\{v_1, \dots, v_l\} \cup \{l+1, \dots, n'\}$ is isomorphic to \mathcal{G}_{out} .

This fact can be proved by a simple induction on l .

Now we can prove the theorem. The resulting graph of Procedure 3 is a graph on the vertex set $\bigcup_i V_i$. According to Fact 2, this graph satisfies properties 1 and 2 of Fact 1. Property 3 of Fact 1 also holds based on Fact 3. Therefore, by the uniqueness of the graph described in Fact 1, we are done. \square

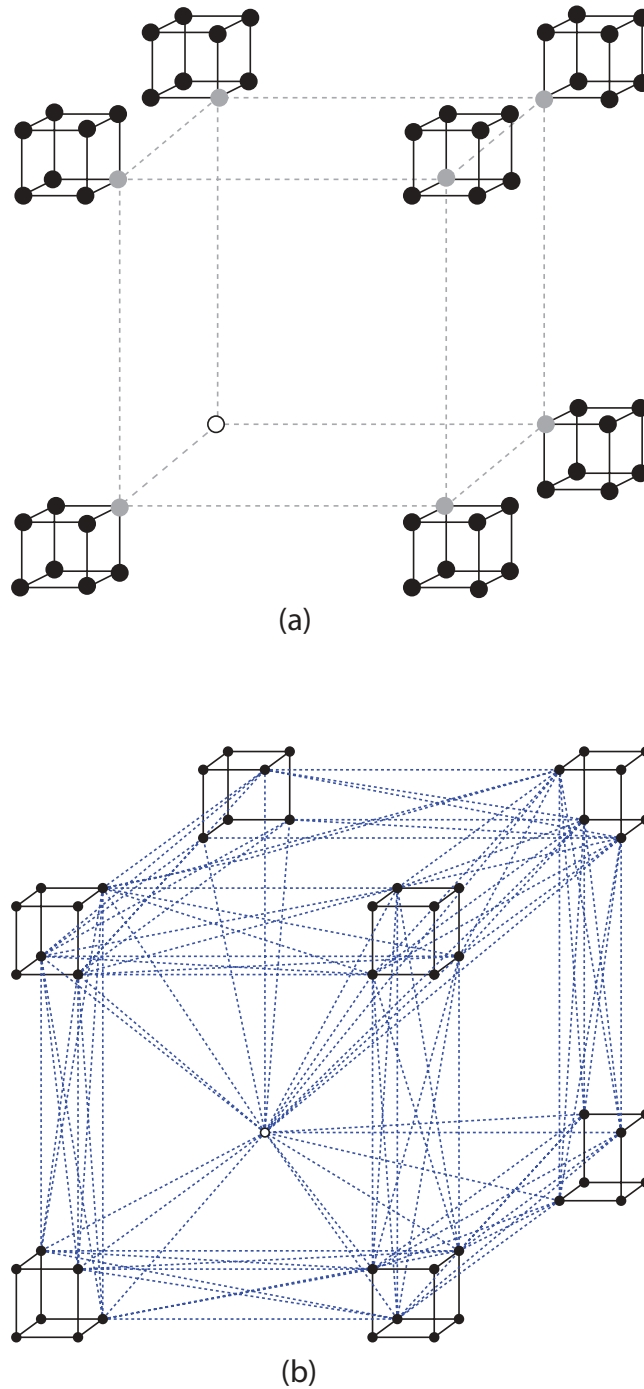


FIG. 8. (Color online) Self-concatenation of Steane's code $[[7, 1, 3]]$ yielding a code $[[49, 1, 9]]$.

We illustrate the graph obtained by generalized local complementation for the code $[[25, 1, 9]]$ which can be obtained by self-concatenation of the code $[[5, 1, 3]]$. As a graph code, the code $[[5, 1, 3]]$ can be described by a pentagon corresponding to the output nodes and a central input node that is connected to all output nodes. Using auxiliary nodes, the concatenated code $[[25, 1, 9]]$ is shown in Fig. 7(a).

The outer code is given by the large pentagon with gray dots and dashed lines. The five copies of the inner code correspond to the small pentagons with black dots and solid lines. The final graph

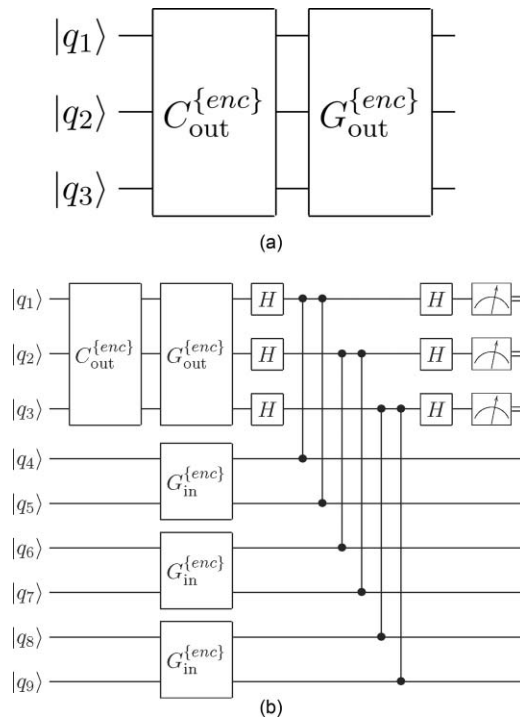


FIG. 9. Encoding circuit for the concatenated code with a general outer code. Here we let $n' = 3$ and $n = 2$. The left circuit illustrates the encoding circuit of the outer code divided into two parts: by $C_{\text{out}}^{\{\text{enc}\}}$ we mean the classical encoder and $G_{\text{out}}^{\{\text{enc}\}}$ is the circuit corresponding to the graph \mathcal{G}_{out} . The right circuit is the encoding circuit of the concatenated code.

is shown in Fig. 7(b). The five solid black pentagons remain, and any vertex in a small pentagon is connected with (blue) dashed lines to any vertex of the neighboring pentagons as well as the central input node.

The situation for the self-concatenation of Steane's code $[[7, 1, 3]]$, which can be realized as a cube, is shown in Fig. 8.

For qubit codes, we see that there is a simple rule to derive the final graph without the auxiliary vertices:

- Any edge connecting an input vertex with an auxiliary vertex is replaced by a set of edges connecting the input vertex with all neighbors of the auxiliary vertex.
- Any edge between two auxiliary vertices A and B is replaced by a complete bipartite graph connecting any neighbor of A with all neighbors of B .

B. A general outer code

In this section, we consider the case when the outer code is nonadditive. The advantage of Theorem 2 is that it directly applies to this case as well.

Procedure 4: (Encoding circuit for \mathcal{Q}_c with a general outer code and an inner code encoding a single qupit)

1. Apply the encoding circuit of \mathcal{Q}_{out} that encodes K' states into n' qupits which we call $q_1, \dots, q_{n'}$.
2. Apply n' copies of the circuit that gives the graph state corresponding to \mathcal{G}_{in} .
3. Apply H^\dagger on all qupits $q_1, \dots, q_{n'}$.
4. Apply the corresponding controlled-Z operators between these qupits and the graph states of \mathcal{G}_{in} .
5. Apply H on $q_1, \dots, q_{n'}$.
6. Measure $q_1, \dots, q_{n'}$ in the computational basis.

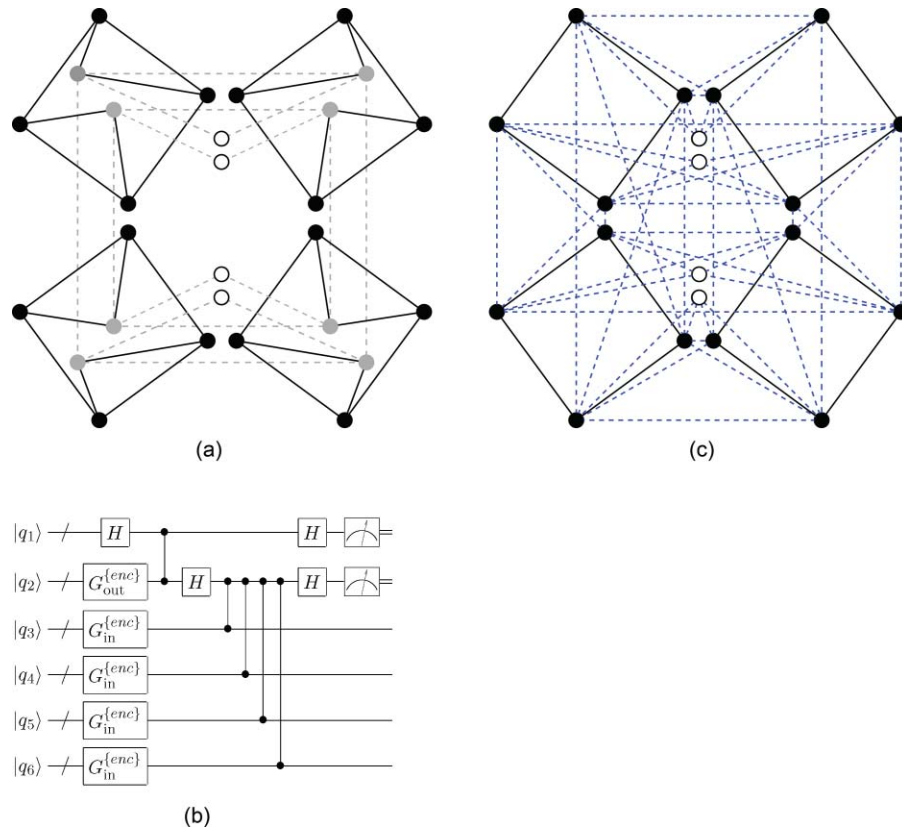


FIG. 10. (Color online) Graphs and encoding circuit for the concatenated $[[16, 4, 4]]_2$ code obtained by concatenating an inner code $[[4, 2, 2]]_2$ with an outer code $[[4, 2, 2]]_2$.

For an example, see Fig. 9.

Note that Theorem 2 deals with steps 3, 4, and 5 in Procedure 4, which are exactly the same as steps 3, 4, and 5 given in Procedure 2. Therefore, whether \mathcal{C}_{out} is linear or not does not actually matter. Consequently, Corollary 1, and thus the main result, holds even for nonlinear outer codes.

C. The case $k > 1$

Theorem 2 can also be directly applied to the case when the inner code encodes more than one qupit. Again, to see this we only need to specify the encoding circuit of \mathcal{Q}_c .

Procedure 5: (Encoding circuit for \mathcal{Q}_c with a general outer code and an inner code encoding k qupits)

1. Apply the encoding circuit of \mathcal{Q}_{out} that encodes K' states (or kk' qupits if \mathcal{C}_{out} is linear) into kn' qupits which we call $q_1, \dots, q_{kn'}$.
2. Apply n' copies of the circuit that gives the graph state corresponding to \mathcal{G}_{in} .
3. Apply H^\dagger on all qupits $q_1, \dots, q_{kn'}$.
4. Apply the corresponding controlled-Z operators between these qupits and the graph states of \mathcal{G}_{in} .
5. Apply H on $q_1, \dots, q_{kn'}$.
6. Measure $q_1, \dots, q_{kn'}$ in the computational basis.

Note that steps 3, 4, and 5 remain the same as those given in Procedure 2. Consequently, Corollary 1, and hence our main result, holds for the case of $k > 1$.

For an example, the left graph of Fig. 10 is the encoding graph $\mathcal{G}_{Q_c}^{C_{out}\{enc\}}$ of the concatenated code Q_c with a $[[4, 2, 2]]_2$ inner code and a $[[4, 2, 2]]_2$ outer code. Note that we decompose the outer code into two copies of a qubit code $[[4, 2, 2]]_2$. Hence there are $kn' = 2 \times 4 = 8$ auxiliary vertices (gray vertices) in $\mathcal{G}_{Q_c}^{C_{out}\{enc\}}$. The corresponding encoding circuit is given by the circuit in Fig. 10, where “/” on each line indicates that there are a set of qubits, not just one. For instance, the line corresponding to $|q_0\rangle$ represents the four input qubits (four white vertices in the left graph of Fig. 10), the line corresponding to $|q_1\rangle$ represents the eight auxiliary qubits, and the line corresponding to $|q_2\rangle$ represents the four output qubits of a single inner code Q_{in} . The graph $\mathcal{G}_c^{C_c}$ of the concatenated code Q_c can be obtained from the encoding graph $\mathcal{G}_{Q_c}^{C_{out}\{enc\}}$ by applying Corollary 1. The result is shown as the right graph in Fig. 10. The (blue) dashed lines are the edges obtained by generalized local complementation.

V. GENERALIZED CONCATENATED CODES

In this section, we discuss the application of our main result to the case of GCQCs. The construction of GCQCs has been recently introduced in Refs. 10 and 11. It resulted in many new QECCs, both stabilizer codes and nonadditive codes.

A GCQC is derived from an inner quantum code $Q_{in}^{(0)} = ((n, q_1 q_2 \cdots q_r, d_1))_p$, which is first partitioned into q_1 mutually orthogonal subcodes $Q_{in\{i_1\}}^{(1)}$ ($0 \leq i_1 \leq q_1 - 1$), where each $Q_{in\{i_1\}}^{(1)}$ is an $((n, q_2 \cdots q_r, d_2))_p$ code. Then each $Q_{in\{i_1\}}^{(1)}$ is partitioned into q_2 mutually orthogonal subcodes $Q_{in\{i_1 i_2\}}^{(2)}$ ($0 \leq i_2 \leq q_2 - 1$), where each $Q_{in\{i_1 i_2\}}^{(2)}$ has parameters $((n, q_3 \cdots q_r, d_3))_p$, and so on. Finally, each $Q_{in\{i_1 i_2 \dots i_{r-2}\}}^{(r-2)}$ is partitioned into q_{r-1} mutually orthogonal subcodes $Q_{in\{i_1 i_2 \dots i_{r-1}\}}^{(r-1)} = ((n, q_r, d_r))_p$.

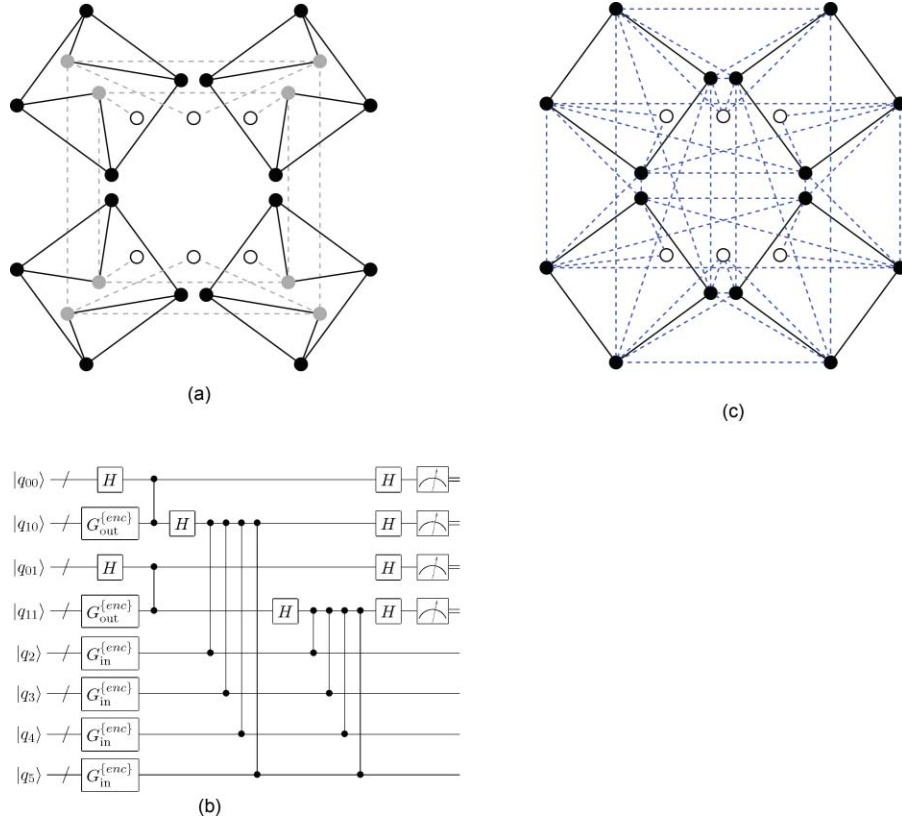


FIG. 11. (Color online) Graphs and encoding circuit for the generalized concatenated $[[16, 6, 2]]_2$ code, derived from an inner code $[[4, 2, 2]]_2$ and outer codes $[[4, 2, 2]]_2$ and $[[4, 4, 1]]_2$.

for $0 \leq i_{r-1} \leq q_{r-1} - 1$, where $d_1 \leq d_2 \leq \dots \leq d_r$, i.e.,

$$9\mathcal{Q}_{\text{in}}^{(0)} = \bigoplus_{i_1=0}^{q_1-1} \mathcal{Q}_{\text{in}\{i_1\}}^{(1)}, \quad \mathcal{Q}_{\text{in}\{i_1\}}^{(1)} = \bigoplus_{i_2=0}^{q_2-1} \mathcal{Q}_{\text{in}\{i_1 i_2\}}^{(2)}, \quad \dots, \quad \mathcal{Q}_{\text{in}\{i_1 i_2 \dots i_{r-2}\}}^{(r-2)} = \bigoplus_{i_{r-1}=0}^{q_{r-1}-1} \mathcal{Q}_{\text{in}\{i_1 i_2 \dots i_{r-1}\}}^{(r-1)}. \quad (52)$$

In addition, we take as outer codes a collection of r quantum codes $\mathcal{Q}_{\text{out}}^{(1)}, \dots, \mathcal{Q}_{\text{out}}^{(r)}$, where $\mathcal{Q}_{\text{out}}^{(j)}$ is an $((n', K'_j, d'_j))_{q_j}$ code over the Hilbert space $\mathcal{H}_{q_j}^{\otimes n'}$.

The generalized concatenated code \mathcal{Q}_{gc} is a quantum code in the Hilbert space $\mathcal{H}_q^{\otimes nn'}$ of dimension $K' = K'_1 K'_2 \dots K'_r$. The detailed construction of \mathcal{Q}_{gc} can be found in Ref. 11. Here we only emphasize that the essence of the “generalization,” which is different from the usual concatenated quantum codes, is that the outer code is actually a product of r outer codes and the inner code is nested-decomposed to specify how that product of outer codes is encoded into each inner code. Therefore, similar to a concatenated code \mathcal{Q}_c , a GCQC \mathcal{Q}_{gc} with a graph inner code

$$\mathcal{Q}_{\text{in}}^{(0)} = (\mathcal{G}_{\text{in}}^{(0)}, \mathcal{C}_{\text{in}}^{(0)}) \quad (53)$$

and r CWS outer codes

$$\mathcal{Q}_{\text{out}}^{(j)} = (\mathcal{G}_{\text{out}}^{(j)}, \mathcal{C}_{\text{out}}^{(j)}) \quad (54)$$

naturally has an encoding graph, denoted by $\mathcal{G}_{\mathcal{Q}_{gc}}^{\{\text{enc}\}}$, and the corresponding encoding circuit is given by the following procedure.

Procedure 6: (Encoding circuit for generalized concatenated code \mathcal{Q}_{gc})

1. Apply the encoding circuits of $\mathcal{Q}_{\text{out}}^{(j)}$ that encodes K'_j states (or $k'_j \log_p q_j$ qubits if $\mathcal{C}_{\text{out}}^{(j)}$ is linear) into $n' \log_p q_j$ qubits which we call $q_1, \dots, q_{n' \log_p q_j}$.
2. Apply n' copies of the circuit that gives the graph state corresponding to \mathcal{G}_{in} .
3. For each $j = 1, \dots, r$, apply H^\dagger on all qubits $q_1, \dots, q_{n' \log_p q_j}$.
4. Apply the corresponding controlled-Z operators between these qubits and the graph states of \mathcal{G}_{in} .
5. For each $j = 1, \dots, r$, apply H on $q_1, \dots, q_{n' \log_p q_j}$.
6. For each $j = 1, \dots, r$, measure $q_1, \dots, q_{n' \log_p q_j}$ in the computational basis.

Note that steps 3, 4, and 5 remain the same as those given in Procedure 2. Consequently, a result similar to that in Corollary 1 holds for constructing GCQCs as well.

Corollary 2: $\mathcal{Q}_{gc} = (\mathcal{G}_{gc}, \mathcal{C}_{gc})$, where \mathcal{G}_{gc} can be obtained for the encoding graph $\mathcal{G}_{\mathcal{Q}_{gc}}^{\{\text{enc}\}}$ via Procedure 3 and \mathcal{C}_{gc} is the classical generalized concatenated code with inner code $\mathcal{C}_{\text{in}}^{(0)}$ (with corresponding decomposition given by the decomposition of $\mathcal{Q}_{\text{in}}^{(0)}$, see Ref. 11 for details) and the outer codes $\mathcal{C}_{\text{out}}^{(j)}$ ($j = 1, \dots, r$).

For an example, the left graph of Fig. 11 is the encoding graph $\mathcal{G}_{\mathcal{Q}_{gc}}^{\{\text{enc}\}}$ of the GCQC \mathcal{Q}_{gc} with a $[[4, 2, 2]]_2$ inner code that is decomposed into two copies of a code $[[4, 1, 2]]_2$. There are two different outer codes $[[4, 4, 1]]_2$ and $[[4, 2, 2]]_2$. Note that there are $4 + 2 = 6$ input vertices (white vertices) and 8 auxiliary vertices (gray vertices). The corresponding encoding circuit is given by the circuit in Fig. 11, where “/” on each line means that the line actually represents a set of qubits. For instance, the line corresponding to $|q_{00}\rangle$ represents the four input qubits of the $[[4, 4, 1]]_2$ outer code, the line corresponding to $|q_{01}\rangle$ represents the two input qubits of the $[[4, 2, 2]]_2$ outer code, the lines corresponding to $|q_{10}\rangle$ and $|q_{11}\rangle$ represents the four auxiliary qubits of the $[[4, 4, 1]]_2$ and $[[4, 2, 2]]_2$ outer codes, respectively, and the line corresponding to $|q_2\rangle$ represents the four output vertices in a single \mathcal{Q}_{in} . To obtain the graph $\mathcal{G}_{gc}^{\mathcal{C}_{gc}}$ of the concatenated code \mathcal{Q}_{gc} from the encoding graph $\mathcal{G}_{\mathcal{Q}_{gc}}^{\{\text{enc}\}}$, apply Corollary 2. The result is shown as the right graph in Fig. 11.

VI. CONCLUSION AND DISCUSSION

In this paper we develop a systematic method for constructing concatenated quantum codes based on graph concatenation, where graphs representing the inner and outer codes are concatenated

via a simple graph operation called generalized local complementation. The outer code is chosen from a large class of quantum codes, called CWS codes, which includes all the stabilizer codes as well as many good nonadditive codes. The inner code is chosen to be a stabilizer code. Despite the restriction that the inner code must be a stabilizer code, our result applies to very general situations—both binary and nonbinary concatenated quantum codes and their generalizations.

Our results indicate that graphs indeed capture the quantum part of the QECCs. Once the graph part is taken care of, the construction of quantum code is reduced to a purely classical problem. This was essentially the idea of the CWS framework (i.e., the problem of constructing a CWS quantum code is reduced to the problem of finding a classical code with error patterns induced by a given graph). Here we have demonstrated that this idea extends to the construction of (generalized) concatenated quantum codes as well [i.e., to construct (generalized) concatenated quantum codes, given the rule of graph concatenation, one only needs to construct the (generalized) classical concatenated codes]. We believe that our results shed light on the further understanding of the role that graphs play in the field of quantum error correction and other related areas in quantum information theory.

VII. ACKNOWLEDGMENTS

We thank Runyao Duan for helpful discussions. S.B. is supported in part by NSF under Grant No. PHY-0803371 and by NSA/ARO under Grant No. W911NF-09-1-0442. B.Z. is supported by NSERC and QuantumWorks. The Centre for Quantum Technologies is a Research Centre of Excellence funded by the Ministry of Education and the National Research Foundation of Singapore.

- ¹M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2000).
- ²D. Gottesman, Ph.D. dissertation, California Institute of Technology, 1997; e-print arXiv:quant-ph/9705052.
- ³A.R. Calderbank, E.M. Rains, P.W. Shor, and N.J. A. Sloane, *IEEE Trans. Inf. Theory* **44**, 1369 (1998).
- ⁴A. Cross, G. Smith, J. Smolin, and B. Zeng, *IEEE Trans. Inf. Theory* **55**, 433 (2009).
- ⁵I. Chuang, A. Cross, G. Smith, J. Smolin, and B. Zeng, *J. Math. Phys.* **50**, 042109 (2009).
- ⁶X. Chen, B. Zeng, and I. Chuang, *Phys. Rev. A* **78**, 062315 (2008).
- ⁷D. Schlingemann, *Quantum Inf. Comput.* **2**(4), 307 (2002).
- ⁸M. Grassl, A. Klappenecker, and M. Rötteler, in *Proceedings of the 2002 IEEE International Symposium on Information Theory*, Lausanne, Switzerland, 30 June–5 July 2002 (IEEE, Piscataway, NJ, 2002), p. 45; e-print arXiv:quant-ph/0703112.
- ⁹D. Schlingemann and R.F. Werner, *Phys. Rev. A* **65**, 012308 (2001).
- ¹⁰M. Grassl, P. Shor, G. Smith, J. Smolin, and B. Zeng, *Phys. Rev. A* **79**, 050306 (2009).
- ¹¹M. Grassl, P. Shor, and B. Zeng, in *Proceedings of the 2009 IEEE International Symposium on Information Theory*, Seoul, Korea, 28 June–3 July 2009 (IEEE, Piscataway, NJ, 2009), p. 953; e-print arXiv:0905.0428.
- ¹²E. Knill and R. Laflamme, e-print arXiv:quant-ph/9608012.
- ¹³E. Knill, R. Laflamme, and W. Zurek, e-print arXiv:quant-ph/9610011; E. Knill, R. Laflamme, and W. Zurek, “Resilient Quantum Computation: Error Models and Thresholds,” *Proc. R. Soc. Lond. A* **454**, 365 (1998), e-print arXiv:quant-ph/9702058.
- ¹⁴C. Zalka, e-print arXiv: quant-ph/9612028.
- ¹⁵D. Aharonov and M. Ben-Or, in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, El Paso, TX, 4–6 May 1997 (ACM, New York, 1997), pp. 176–188; e-print arXiv:quant-ph/9611025.
- ¹⁶M. Hein, W. Dur, J. Eisert, R. Raussendorf, M. Van den Nest, and H.J. Briegel, in *Proceedings of the International School of Physics “Enrico Fermi” on “Quantum Computers, Algorithms and Chaos,” Varenna, Italy, July 2005* (IOS Press, Amsterdam, The Netherlands, 2006); e-print arXiv:quant-ph/0602096.
- ¹⁷M. Bahrngiri and S. Beigi, e-print arXiv:quant-ph/0610267.
- ¹⁸J. Dehaene and B. De Moor, *Phys. Rev. A* **68**, 042318 (2003).
- ¹⁹J. Dehaene, E. Hostens, and B. De Moor, *Phys. Rev. A* **71**, 042315 (2005).
- ²⁰J. Dehaene, M. Van den Nest, and B. De Moor, *Phys. Rev. A* **69**, 022316 (2004).